IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:	
CRAIG L. SCHIMMEL	
Serial No.: 10/797,369	Art Unit: 2676
Filed: March 9, 2004	Examiner: RAHMJOO, Manucher
For: REAL TIME CONTOUR LINE GENERATION	

AFFIDAVIT OF CRAIG L. SCHIMMEL

1. My name is Craig L. Schimmel and I am an inventor in the above-referenced patent application.

2. Attached is a copy of my curriculum vitae.

3. I have been a software engineer for ten years. I have been working on avionics display processing for six years at Honeywell International, Inc. Of that, three years have included work on real time digital maps. My three years of digital map work experience includes R&D on the techniques and technologies used in the Honeywell embedded digital map systems. This patent application is a result of that R&D work. Due to my education and experience, I consider myself an expert in the art.

4. I have reviewed the office action regarding the pending patent application dated May 25, 2005. I strongly disagree with the Examiner's statement that Beckwith teaches a "state" and "next state" for his digital map generator and display system. In addition I strongly disagree that Beckwith discusses using base row and column elevation values and updated row and elevation vales as specifically claimed in the present invention. Beckwith teaches **[need to provide]**

5. Beckwith does not teach the feature of *setting row and column data to an initial contour value, the row and column comprising a first state.* The contour table referred to by Examiner in Beckwith column 17 rows 54-55 contains *preprocessed* information. To quote, Beckwith column 17 rows 24-34 "A contour table is introduced ... containing pertinent shades of gray and contour line data as *preprocessed* information..." (emphasis added). Preprocessed data does not represent a state, to be updated during processing of source elevation data. Rather, as described in Beckwith column 17 rows 29-33 it is a table which is indexed into based on the incoming source elevation data. It is not an updated processing state (next row and column data), as is specifically described and claimed in the present invention.

Additionally, Beckwith column 17 rows 45-50 clearly state that the contour table values are only updated when new visual requirements are selected. That is, when a different contour interval or shading value is selected by the external user. This differs greatly from a processing state as described in the present invention. In the present invention, the row and column data represent the processing state which is updated during the course of processing a set of elevation data. This state is given an initial value which represents a first state, where the state will change during the course of processing a set of elevation data. The table described in Beckwith, as demonstrated above, is given a value which does not change during the course of processing, and is therefore not a state, nor an updated state, with regard to the processing of the elevation data set.

6. Beckwith does not teach the step of *comparing a second data point with the first state for determining an existence of a contour line depending on a result from the step of comparing*. Beckwith at column 17 rows 3-5 compares two adjacent *data points,* "Are any two adjacent data points located in different contour intervals?" In this instance, Beckwith is teaching the comparison of neighboring data points. This differs from the present invention, which compares a data point with the current processing state, where the processing state is identified as the row and column data. The state contained in the row and column of the present

invention should not be confused with the input elevation data itself, which happens to be arranged in a row/column format. As stated in claim 1 c), the present invention compares a data point, from the input elevation data, with the processing state to determine the presence of a contour line. Furthermore, the process of Beckwith "Are any two adjacent points ..." necessarily requires the comparison of a data point with its 4 neighbors. Each interior data point has 4 neighbors (above, below, left, right); therefore comparison of adjacencies requires 4 comparisons. Beckwith states this in column 16 rows 49-50. This contrasts with the present invention which compares a data point only with the current state, not with adjacent data points, as stated in claim 1 c). Therefore, the present invention's method of comparing a data point against a first state differs from Beckwith's comparison of a data point with adjacent data points.

7. Beckwith does not teach the step of *updating the first state to a next state, wherein the next state comprises a next row and column data if the contour line exists.* The rejection refers to Beckwith column 17 rows 15-21. The referenced section of Beckwith describes generating a signal for the purposes of displaying a portion of the contour line on the output device. The method used in Beckwith to display a point once a contour has been found is moot with regard to the process of determining if a contour exists. The referenced section does not state or suggest that any processing state has been changed. In contrast, the present invention maintains and updates the processing state as a method of determining if a contour line exists. The state maintained in the present invention is named the row and column data in claim 1, and contains the current elevation level achieved. From claim 1 d), this processing state is updated every time a contour line is found. Beckwith in column 17 rows 38-40 also offered as evidence of Beckwith teaching the storing of a next state. This section of Beckwith clearly states the contour table "*outputs*", not the contour table itself, are updated and stored. That is, after determining that a contour exists, Beckwith stores the result (is there a contour). This differs from the present invention, which stores the processing state, which is used to determine if there is a contour line.

8. I am familiar with several publications and other experts in similar art areas, and they would also vigorously disagree with this contention made by the Examiner.

Further, Affiant sayeth naught.

STATE OF NEW MEXICO)
)
COUNTY OF BERNALILLO)

IN TESTIMONY WHEREOF, Craig L. Schimmel has hereunto set his hand this 12 day of July, 2005.

SIGNATURE: Craig L. Schimmel

SUBSCRIBED AND SWORN to before me this 12 day of July 2005, by Craig L. Schimmel.

Virginia L. Larsen
NOTARY PUBLIC

My Commission Expires:
4-22-2008

OFFICIAL SEAL
Virginia L. Larsen
NOTARY PUBLIC
STATE OF NEW MEXICO
My Commission Expires: 4-22-2008

Craig L. Schimmel

Summary

A software engineer with 10 years of experience developing graphical software for embedded systems and workstations. Strengths include software design, C and C++ programming, real time embedded systems OpenGL, and UNIX.

Experience

Honeywell International, Defense Avionics, Albuquerque NM
1999 to present

Principal Software Engineer

Software Team Lead for the Digital Map Processor

- Lead a team of 5 software engineers through the entire development life cycle.
- Responsible for software team performance, software architecture, system integration & test, and customer interaction.
- Provide technical and professional mentoring to junior team members.
- Developed the software for a multiprocessor real time embedded digital map system utilizing VxWorks, C, and OpenGL.
- Developed VxWorks BSP for custom hardware, including development of drivers for SCSI, graphics processor, system controller and FPGA.
- Developed UNIX tools and utilities using C++ and Perl
- Utilized ISO 9001 and CMM level 3 practices.

Lead software engineer for digital map IR&D project

- Technical design resulted in a patent application and a technical achievement award.
- Produced the core design for future Honeywell digital map systems, to be utilized on multiple projects.
- Enhanced OpenGL drivers, particularly texture functionality.
- Modified VxWorks BSP functions.

Division expert for graphics processing and real time embedded system issues.

- Lead software engineer on multiple high priority tiger teams, formed to resolve critical technical issues.

- Called on to diagnose, debug and resolve technical issues for teams throughout the division.

Senior Software Engineer

Software engineer for the F-16 embedded graphics processing

- Full life-cycle development.
- Developed the real-time OpenGL graphics software application.
- Modified VxWorks BSP and graphics drivers.
- Created a distributed build system, code generators, utilities, and test tools using make, C++, PERL, and Java.
- Utilized diagnostic tools including oscilloscopes, logic analyzers and software performance monitors.
- Created Microsoft Windows based diagnostic and production tools.

Raytheon, Missile Systems, Tucson AZ
1994 to 1999

Software Engineer

Software engineer for a real time, distributed simulation environment.

- Interconnected systems, sensors, simulations and displays dispersed across a large geographical area in real time.
- Developed cross-platform network infrastructure using CORBA and DoD RTI middle-ware. Platforms included VxWorks, UNIX and Microsoft Windows.
- Developed GUI data gathering and analysis tools using C++, Tcl/Tk, and GTK.
- Administered lab network of Sun and SGI workstations.
- Provided technical presentations to customers and colleagues.

Simulation and visualization software engineer

- Maintained simulation and visualization tools using C, C++ and OpenGL.
- Maintained FORTRAN and C based simulations.
- Developed data analysis and GUI display tools using C, PERL, and the GTK X Windows toolkit.
- Interfaced legacy simulations to a distributed simulation network using C++ and TCP/IP.
- Developed 3D models using MultiGen modeling tools.
- Maintained software documentation and configuration management.

Awards and Publications

Honeywell Technical Achievement Award, 2004
Honeywell Outstanding Engineer Award, 2003
"Design and Development of Standard Missile Block III System Test Bed Using HLA", co-author, Presented at SISO conference 1999

Education and Training

B.S. Computer Science and Engineering, Northern Arizona University	1994
PowerPC 74XX Design Training,	2004
ACM Online Java Courseware,	2003
Wind River Tornado BSP Training,	2001
Wind River Tornado Training,	1999
MultiGen Modeling Training,	1996

Professional

Member ACM and SIGGRAPH

Programming Languages

Skilled: C, C++, Objective-C , PowerPC assembly, PERL, shell scripts, FORTRAN

Experienced: Java, Python, Lisp, X86 assembly

Operating Systems

VxWorks, UNIX (Solaris, Linux, IRIX), Mac OS X, MS Windows

Other Technical Skills

OpenGL, 2D/3D graphics, embedded real time systems, display systems, digital maps, distributed systems, device drivers, BSP, computer simulation, networking, TCP/IP, CVS, procedural and object oriented design, GUI toolkits, makefiles, HTML/XML, LightWave